Combining knowledge discovery, ontologies, annotations, and semantic wikis

Hernan Astudillo, Victor Codocedo, Gérôme Canals, Diego Torres, Alicia Diaz, Amedeo Napoli, Alan K. Gomes, Maria-Graça Pimentel

Chapter 6

Combining knowledge discovery, ontologies, annotations, and semantic wikis

Hernán Astudillo, Víctor Codocedo, Gérôme Canals, Diego Torres, Alicia Diaz, Amedeo Napoli, Alan K. Gomes, Maria Graça C. Pimentel

Abstract

Semantic Wikis provide an original and operational infrastructure for efficiently combining semantic technologies and collaborative design activities. This text presents: a running example and its context (organization of the collections in a museum); concepts of wikis as a tool to allow computer supported cooperative work (cscw); concepts of semantic technologies and knowledge representation; concepts and examples of semantic wikis; anatomy of a semantic wiki (reasoning tools, storage, querying); and research directions.

6.1. Introduction

The evolution of the Web has demanded research efforts in several areas, from support to building applications [12] to the evaluation of user interfaces [18]; from investigating the application of information retrieval in general [21] to the building of efficient metasearch engines [27], web clustering engines [10], and machine learning techniques for web text categorization [33].

Computer supported cooperative work (CSCW) was identified in the early 80's as an important research area, which demanded investigation of alternatives to support collaborative activities in several areas [13] [41]. CSCW investigates the understanding of the way people work in groups with the support of information technologies [8]. Collaborative work on the Web was made possible via the asynchronous distributed collaborative authoring of web pages allowed by the early wikis proposed by Cunningham[1].

A Wiki is roughly speaking a Web site (built around a set of pages) that can be edited by several people, possibly at the same time. Editing is the collaborative task and problems of coherence and needs for synchronization appear while several people are working together.

[1]http://www.c2.com/cgi/wiki?WikiWikiWeb

The provision of alternatives to associate metadata with the information in wikis is an important research topic [15], and the association of ontologies a natural alternative [37].

Combining wikis and semantic Web technologies is considered as a promising alternative for collaboratively creating and using information on the Web. The user-friendliness of wikis as regarding multi-site content generation and the power of semantic technologies for organizing and retrieving knowledge may complement one another towards a new generation of Web-based content management systems.

Accordingly, a semantic wiki can be seen as a wiki including an associated ontology, i.e. an operational representation model of domain knowledge, that can be used for annotating the content of Wiki pages and used for typing hyperlinks and testing consistency of contents (e.g. two users cannot state contradictory facts with respect to the underlying ontology). Moreover, an annotation process with respect to an ontology has a direct impact on knowledge access, semantic search and reuse, collaborative authoring, and social collaboration [3] [22].

The remaining of the text is organized as follows. In the Section 6.2 we present an example of a semantic wiki in the Museums and Archives domain. In Section 6.3 we discuss the association of knowledge discovery and representation with ontologies and semantic wikis. We detail annotation aspects of Semantic Wikis in Section 6.4, and implementation aspects of Semantic Wikis in Section 6.5. We point to recent research directions in Section 6.6.

6.2. Semantic Wiki and Museums: An example

Museums posses a great amount of knowledge about the context of the documents and art works that they host. A proper documentation of such knowledge not only enriches the visitor experience in the museum but also improves research and allows a better understanding of the document. This section provides an example of the benefits on using a semantic wiki in the Museums and Archives domain.

6.2.1. Museums Online - Current Reality

Most of major Museums in the world have their own online catalogs providing a brief summary of the information of their documents (documents refer to any kind of art work that Museums or Archive could host: Paintings, Pictures, Sound Records, etc).

Figure 6.1 shows information about “La Gioconda” currently exhibited on Louvre Museum on France and extracted from their online catalog [24]. It contains a digital representation of the document and provides information about the author, the years in which it was probably painted, a description of what represents (a portrait of Lisa Gherardini) and a description of the history of the painting in the form of a large text.

We define “Context” or “Background Knowledge of a document” as the kind of data that put a frame of reference on a document and, more importantly, allows to connect several documents through a set of dimensions (temporal dimension, spatial dimension, style dimension, etc). In the case of the information about “La Gioconda” the history of the painting represents a large amount of background knowledge available. Unfortunately,

Figure 6.1. Background knowledge provided by Louvre about "La Gioconda"

this it's not always the case.

Figure 6.2 is the information provided by the online catalog of the "Metropolitan Museum of New York" of "The Death of Leonardo da Vinci" by Giuseppe Cades [28]. For this document there's a digital representation, information about the years when it was probably painted and other physical descriptions but as it can be seen, information about the history of the art work it is not provided as in La Gioconda's case.

6.2.2. Who is Leonardo da Vinci?

Because of the fame of "La Gioconda" it's clear to us that its author is Leonardo da Vinci. But if we look at Figure 6.1, in the information provided by Louvre about "La Gioconda" we find that its author is "Leonardo di ser Piero". The fact is that the actual name of Leonardo da Vinci is "Leonardo di ser Piero" — da Vinci it's just to make clear that he was born at the Tuscan town of Vinci.

If we think about a system that could integrate the catalogs of Louvre and the MET Museum, there should be a way to inform the system that "Leonardo da Vinci" is an alias of Leonardo di ser Piero: this should allow the the system to compute that the author of "La Gioconda" is the same person painted on "The Death of Leonardo da Vinci". It should be also appropriate to inform the system that "Leonardo di ser Piero" is an Italian person, who was a painter, architect, musician, inventor, engineer and sculptor, who was born on 1452 and died on 1519. This background knowledge complements each document where Leonardo da Vinci is present (as an author, as a content, as an inspiration for another author, etc), and provides and integrated base for every other museum.

6.2.3. A Semantic Wiki for Museums

In this section we present an illustrative implementation of the Semantic Media Wiki [34] using as content some metadata associated with documents from Louvre and MET museums, as well as information from the Wikipedia. We show the benefits of using this

Figure 6.2. Data of "The Death of Leonardo da Vinci" on MET Museum

approach and how it can be used to integrate museum's online catalogs. We also discuss how this approach can help resource-less museums and may enable a better interaction between museums and their visitors.

6.2.3.1. Background Knowledge from Wikipedia

Figure 6.3 presents an extract of the Semantic Wiki page of Leonardo da Vinci created from information taken from Wikipedia [42].

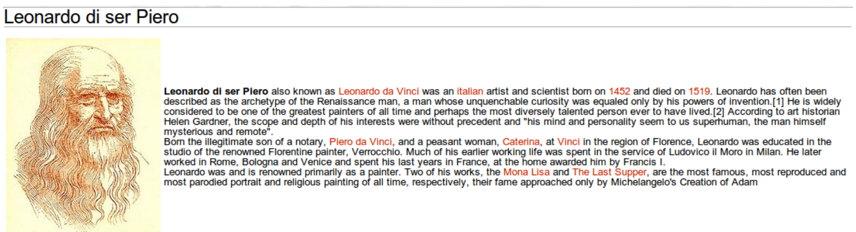


Figure 6.3. Semantic Wiki Page of "Leonardo di Ser Piero" better known as Leonardo da Vinci

Figure 6.4 shows an annotation inserted in the Semantic Wiki page: the annotation specifies data and metadata associated with Leonardo's year of birth, year of death, nationality and alias.

For this example, we have created 10 pages in the Semantic Wiki: the title of the painting (work), the name of the painter (author), the location of the painting (Museum) are listed in Figure 6.5).

```
[[name::Leonardo di ser Piero]]
  also known as [[alias::Leonardo da Vinci]]
  was an [[nationality::italian]] [[profession::painter | artist]]
  and scientist
  born on [[year of birth::1452]]
  and died on [[year of death::1519]].
```

Figure 6.4. Extract of the Semantic Media Wiki annotation of Leonardo da Vinci's wiki page

Art Work Title	Author	Museum
Coronation of Napoleon	Jacques-Louis David	Louvre
Gardanne	Paul Cézanne	MET
La Gioconda	Leonardo di ser Piero	Louvre
Mrs. Winthrop W. Aldrich	Joaquín Sorolla y Bastida	MET
Strawberries	Édouard Manet	MET
The Crucifixion	Jacopo Palma the Younger	MET
The Raft of Medusa	Théodore Géricault	Louvre
The Slave	Michelangelo	Louvre
The Wedding Feast at Cana	Paolo Veronese	Louvre
Tobias	Bernardo Strozzi	MET

Figure 6.5. List of 10 documents created in the Semantic Wiki

6.2.3.2. Art work from Museums

Figure 6.6 shows the Semantic Wiki page of the document "La Gioconda" with information extracted from the Louvre.

Figure 6.7 shows an extract of the semantic annotations used for the same Semantic Wiki page. For every document in the Semantic Media Wiki, we have used a semantic template defined in the Semantic Media Wiki.

Along with La Gioconda wiki page, we have created 10 Semantic Wiki pages of documents extracted from the Louvre (5 documents) and the MET online catalogs (5 documents). Figure 6.5 shows the names of the documents, which correspond to the `name` of the painting, and the name of the `author`, which corresponds to the name of the painter in the infobox document.

6.2.4. Exploiting Semantics - MuSem Wiki

In this section we list common tasks that can be performed using the previously described system based on Semantic Media Wiki, MuSem Wiki (Museum on Semantic Wiki).

6.2.4.1. Listing Documents

On a regular wiki, the contents of most pages is created statically, so that if we create a list of the art works on the wiki, for every new document added to the system, we should modify the wiki page wich contains the list. On a Semantic Wiki this is not the case. Since every document is a semantic entity defined by an identifier (URI) along with

La Gioconda

La Gioconda

Name	La Gioconda
URL	Archivo:gioconda.jpg
Alias	Mona Lisa
Author	Leonardo di ser Piero
Year of creation	1503 – 1506
Museum	Louvre
Description	Acquired by Francis I in 1518, acclaimed by artists of the day, the Mona Lisa – also known as La Gioconda – only earned her worldwide fame in the 20th century, more on account of her "adventures" theft (1911–14), stoning (1956), travels to the United States (1963) and Tokyo and Moscow (1974) — than her outstanding qualities. Da Vinci's dazzling, almost magical technique models the forms through his use of glazes (very diluted, quasi-transparent layers of paint), playing with light and shade effects by making the contours hazy ("sfumato"). Aerial perspective, moving from brown to blue, creates, through the density of the air, an abstract landscape made up of earth and water. What a pity that the colors darken as the varnish ages: the sleeves were once saffron yellow. The model's identity has given rise to the oddest suggestions at times, even going as far as to say that she was a man. It is probably a portrait, begun in Florence between 1503 and 1507, of Monna ("Mrs.") Lisa Gherardini del Giocondo. Her smile could thus be a symbol of her name, "gioconda" also meaning "cheerful." While this is one of the period's largest portraits, painted on a single, very thin (12 mm) poplar board, it is not an ostentatious image of a rich bourgeoise lady, although her pose and attire and the absence of eyelashes and eyebrows are in keeping with the elegance of her station. It is above all an ideal portrait, reflecting Renaissance interest in Platonic theory, when the beauty of the body was seen as that of the soul.

Figure 6.6. Semantic Wiki Page of “La Gioconda”

```
{{Infobox document
| name            = La Gioconda
| dig_rep_url = gioconda.png
| alias = Mona Lisa
| author        = Leonardo di ser Piero
| creation_interval = 1503 - 1506
| exh_place = Louvre
| desc = Acquired by Francis ...
}}
```

Figure 6.7. Extract of the Semantic Media Wiki infobox of La Gioconda wiki page

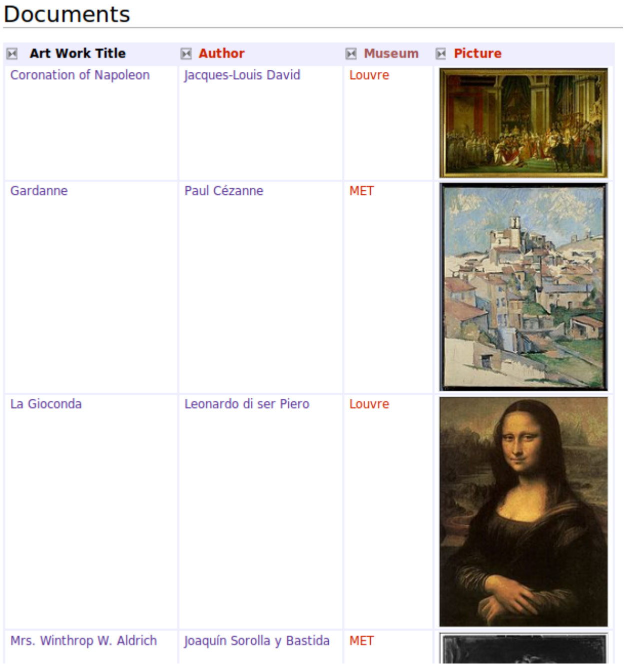
several attributes, it is possible to represent documents in terms of queries to the system. Figure 6.8 shows the query for listing all documents in the system. The first line is the specification of the attributes required for the entities to appear in the list: in the example shown in the figure, the exhibition place is the Louvre `or` the MET. In the following lines we specify the attributes that we want to select for presentation in the wiki page. Figure 6.9 presents a snapshot of the output for the query in Figure 6.8.

6.2.4.2. Listing Documents by kind of Author

Documents can also be listed using properties associated with authors. It is possible, for instance, to create three separate lists of documents categorizing author by their nationality (Italian, French or Spanish). As an example, Figure 6.10 shows the query used for obtaining the list of paintings from French authors. Figure6.11 shows the output of this

```
{{#ask: [[exh_place::MET]] OR [[exh_place::Louvre]]
 | mainlabel = Art Work Title
 | ?author = Author
 | ?exh_place = Museum
 | ?url = Picture
}}
```

Figure 6.8. Query for listing all documents in the system


Documents

Art Work Title	Author	Museum	Picture
Coronation of Napoleon	Jacques-Louis David	Louvre	
Gardanne	Paul Cézanne	MET	
La Gioconda	Leonardo di ser Piero	Louvre	
Mrs. Winthrop W. Aldrich	Joaquín Sorolla y Bastida	MET	

Figure 6.9. List of documents in the system. The name, the author, the museum where it's hosted and a picture are displayed

query. Other lists can be created by changing the value of the nationality in the first line of the query.

6.2.4.3. Propagation of Edits

As it is the case in most wikis, moderators rule the history of changes on the Semantic Wiki pages. In this case, the staff on the museums would have moderator privileges on their own contents so that they could perform changes — when a different (or better) digital representation of the art work is available, for instancce. This change would be propagated to the lists shown on Figure 6.9 and Figure 6.11 without any additional work as those lists are created dynamically from data provided on the wiki pages.

6.2.4.4. Adding a new Museum

Adding new content from another museums gets easy when the system grows as background knowledge already exists for some artists. The Art Museum of Sao Paulo [25]

```
{{#ask: [[author.nationality::french]]
 | mainlabel = Art Work Title
 | ?author = Author
 | ?exh_place = Museum
 | ?url = Picture
}}
```

Figure 6.10. Query for listing art works from French authors

French works

Art Work Title	Author	Museum	Picture
Coronation of Napoleon	Jacques-Louis David	Louvre	
Gardanne	Paul Cézanne	MET	
Strawberries	Édouard Manet	MET	
The Raft of Medusa	Théodore Géricault	Louvre	

Figure 6.11. List of art works from French authors

(MASP) provides a poor on line catalog of its collection (only the title of the painting and the name of the artist). Using a template we can add the document "Allegory of Wisdom and Strength: The Choice of Hercules or Hercules and Omphale" by Paolo Cagliari (already in the system).

Using the template for author, what is important is to be able connect the new document to the document already existing in the system, which specifies the author as Paolo Cagliari (Paolo Veronese's true name). Figure 6.12 shows the wiki code for the new content.

As it can be seen, there are three new properties added for this document. This is because this sample is a copy of the original and was made on 1750 by François Boucher. New properties can be added to the system freely as the Semantic Media Wiki does not restrict nor the content nor the definition of the ontology that supports it. Beause it can provide great flexibility it's important to maintain a coherent ontology so creation of new properties should be restricted to moderators and administrators.

```
{{Infobox document
| name        = Allegory of Wisdom and Strength: The Choice
| dig_rep_url = hercules_and_omphale.png
| alias       = Hercules and Omphale
| author      = Paolo Veronese
| creation_interval = 1576 - 1584
| exh_place   = MASP
| type        = copy
| executer    = François Boucher
| reproduction_year = 1750
| desc        = Not Available
}}
```

Figure 6.12. Wiki code for adding an art work from a new museum

6.2.5. Discussion

In this section we have presented an example implementation of a Semantic Wiki using the Museum's domain. For this we have used some documents (art work information) from Louvre, Metropolitan Museum of New York and Art Museum of Sao Paulo. To create background knowledge about the artists of this art works we have used Wikipedia information. The system, that we have called MuSem Wiki, was implemented using Semantic Media Wiki as a framework.

We have shown that Semantic Wiki pages allow us to create dynamic content as lists and filtered lists. Semantic data on wiki pages also allow us to connect documents from different Museums using background knowledge, and to manage complex features such as aliases and copies.

As MuSem Wiki provides data in a shareable and structured manner, it allows resource-less Museums to use free background knowledge for their documents and to connect their content to other collections in a easy and inexpensive way.

Background knowledge can also be added by people outside the institutions as they can provide data that the museum's researchers could not have (e.g, a descendant of the artist or, for moderns collections, the artists themselves). Connecting Museums and their visitors in such way adds good value for both parties.

Finally, we are confident that this small example shows that great benefits are created using a simple application based on Semantic Wikis. We have not covered the full spectrum of the technical details of Semantic Media Wiki, for further information we recommend the reader to refer Semantic Media Wiki Documentation [34].

6.3. Knowledge discovery, ontologies, and semantic wikis

In this section we discuss the association of knowledge discovery and representation with ontologies and semantic wikis.

6.3.1. Knowledge discovery and representation: from resources to an ontology

Ontologies are the backbone of semantic Web in allowing software and human agents to communicate and to share domain knowledge [1] [35].

From a formal point of view, an ontology is considered as an explicit specification of a domain conceptualization. For being operational, an ontology has to be encoded within a knowledge representation language such as a description logic [5] or OWL [26]. From a practical point of view, besides ontologies, there exist different types of "ontological resources" such as thesaurus, vocabularies, dictionaries, collections of documents, and databases.

Every ontological resource provides a specific aspect of domain knowledge. For taking into account these ontological resources, a framework can be designed in which the content of resources can be integrated for being used as a "domain knowledge container" for knowledge sharing and reuse.

Following this way, Formal Concept Analysis (FCA) and its extension Relational Concept Analysis (RCA) can be used for designing and completing an ontology from resources (especially textual documents as detailed by Bendaoud et al. [6] [5]). FCA and RCA support the transformation process between ontological resources and a concept lattice supporting an "ontology schema". One way of processing is to build a "source ontology" from resources, and then to extend this source ontology by progressively adding new units extracted from additional resources. The transformation process can be based on FCA operations (e.g. apposition) and on RCA operations (e.g. extracting relations between concepts). Finally, elements in the resulting concept lattice can be represented within a knowledge representation language such as OWL for obtaining a concept hierarchy that can be considered as an ontology schema.

In this way, FCA and RCA are "core processes" in the design of a target ontology from a set of heterogeneous resources. Firstly, FCA and RCA take into account all basic elements within an ontology, such as objects (or individuals), attributes, and relations. Secondly, FCA provides operations for creating, managing, and updating concept lattices. When the resulting concept lattices are transformed into concept hierarchies, a classifier can be used for classification-based reasoning, content-based information retrieval, and answering queries.

This is an operational way of designing ontologies from a set of resources but the process is hardly automated without any human intervention. This calls for collaborative architecture and a Wiki provides a convenient infrastructure in which the transformation process from resources to knowledge can be carried out, as explained in the next section.

6.3.2. Wikis and Semantic Wikis

A Wiki is a collection of Web pages allowing collaboration and user-contributed "knowledge production", by enabling users to contribute or modify content of pages [23]. One of the best-known repository of user-contributed knowledge is Wikipedia, also one of the largest and fastest growing online sources of encyclopedic knowledge [39][40]. The richness of its embedded structural information is mainly based on hyperlinks, with relations such as synonymy, polysemy, and additional tools such as infoboxes and templates. A

Wiki offers simplicity and a social dimension, and emergence of structured knowledge repositories of collaborative nature.

Here, knowledge has to be understood as Şknowledge for human agentsŤ and not for software agents. This is why, for guiding a coherent development of a Wiki infrastructure, semantic technologies and especially ontologies can be used within Wikis, leading to Semantic Wikis [3] [22] [9].

Semantic Wikis allow knowledge processing for humans and machines. The Wiki infrastructure can be used for supporting an ontology: the Wiki can be considered as a front-end of the ontology maintenance system, with Wiki pages as concepts, typed links as relations and attributes. By contrast, ontologies can be used within Wikis for supporting page selection, annotation (tag organization), searching, querying, faceted navigation, guided editing, and consistency checking. For example, the Semantic Mediawiki system addresses consistency of contents, knowledge access, and reusing information [22]. Moreover, an annotation process with respect to an ontology has a direct impact on querying information: annotations can be categories, relations, and attributes, and can be represented as logical statements manipulated within reasoning schemes. In the same way, three main tasks linked to Wikis are improved by the presence of an ontology, namely collaborative authoring (editing), social collaboration (change tracking), and semantic search (browsing) [3].

6.3.3. Combining knowledge discovery, ontologies, annotation, and semantic Wikis

There are two main views relating Wikis and ontologies: (i)Wikis for improving ontology infrastructure, and (ii) ontologies for improving the development and the management of Wikis.

- In the first view, collaboration plays an important role for editing pages and for gathering and integrating resources of different types. Knowledge discovery (KD) processes can be applied to such a container of resources for extracting units. These units, after interpretation by an analyst, can be embedded within an ontology. In this view, a semantic Wiki can be used for selecting, collecting, and preparing data (documents) in a collaborative way for ontology design and extension.
- In the second view, an ontology plays the role of a domain model, providing a Şdomain terminologyŤ with terms and associated meaning. In this view, improvements for the Wiki activity are mainly based on document searching Űsearching by contentŰ and understanding, coherence checking, and guided editing. Both views involve collaborative aspects and Şknowledge productionŤ (i.e. extraction or creation): collaboration based on a Wiki guides knowledge organization and evolution in the first view while an ontology can be used at two for controlling the evolution and checking the consistency of the new elements brought through collaboration in the second view.

A Semantic Wiki can be used as a support for knowledge discovery and management combining two interrelated views of Wikis and ontologies. For example, in the context of the design of a course, the following loop summarizes the operations that illustrate this approach:

- Based on an initial ontology, a set of documents of interest ŞrankedŤ by their content with respect to a given topic is selected.
- Then, documents are annotated and related (through hyperlinks) in a collaborative way and under the control of the ontology, i.e. using terms defined within the ontology.
- The resulting set of annotated documents can be analyzed using data mining algorithms for extracting elements of interest.
- These new elements can be interpreted and then proposed as knowledge units for extending and improving the initial ontology. The Wiki infrastructure can be used for making easier interactions.
- The loop is closed: starting from an ontology and going back to the ontology through a collaborative activity within a semantic Wiki using knowledge discovery and knowledge representation techniques.

6.4. Semantic Wikis: Semantic Meets Wikis or Wikis Meets Semantic?

In the last few years, the combination of Social Software with Semantic Web technology has been gaining significant attention in the Semantic Web community. Workshops like the International Workshop in Semantic Wikis at the European Semantic Web Conference, which, since 2006, every year has a high number of submissions and attendees, are significant examples of this interest.

Social software covers a range of software systems that allow users to interact and share data. This computer-mediated communication has become very popular with social sites like *MySpace* and *Facebook*, media sites like *Flickr* and *YouTube*, as well as wiki sites like *Wikipedia*. The terms Web 2.0 are also used to describe this style of software[2]. Although this systems are characterized by the huge amounts of content available, what actually makes them interesting is they considerably change the way the content is created and consumed. In Social Software, users leave to be merely consumers of content to become producer of it.

On the other hand, the vision of the *Semantic Web* is to move from content that is suitable for presentation only to “smart” content that may be processed by machines and used in different settings. It is also to move from application-centric systems to data-centric systems, and from a Web focused on information to a Web focused on relations between things. The current Semantic Web enriches the existing Web with meta-data and (meta-)data processing in order to extend Web-based systems with advanced (intelligent) capabilities. Semantic Web assumes a distributed but strongly connected web of small pieces of formal knowledge rather than big, centralized knowledge bases.

Semantic Social Software is the combination of Social Software with Semantic Web technologies. It is based on Social Software and Semantic Web technologies are remarkably similar. Meanwhile Social Softwaret'content can be see as small but strongly connected pieces of content from different sources with differing opinions, the Semantic

[2]http://en.wikipedia.org/wiki/Social_software

Web deals with small but strongly connected pieces of formal knowledge from different sources, with different levels of precision and trustworthiness, even inconsistencies. The difference between them is only in the level of abstraction: Social Software mostly deals with social connections and human readable content, the Semantic Web mostly deals with formal connections and formal content.

The two basic ideas of Semantic Social Software are: to improve usage of Social Software by adding metadata and to improve the process of creating Semantic Web metadata by using Social Software. These ideas are in concordance with Nova Spivackt's MetaWeb [3] approach, which is essentially about using social connections to form information connections and vice versa. Nova Spivack is convinced that "The Metaweb is emerging from the convergence of the Web, Social Software and the Semantic Web," connecting human and machine intelligence and moving from "just a bunch of interacting parts" to "a new higher-order whole". Schaffert [30] describes two different perspectives to conceive the Semantic Social Software, namely

Semantically Enabled Social Software It makes use of semantic metadata to enhance existing social software. As it was pointed above, massive amounts of digital content, which is connected via hyperlinks and/or social networks, are available in Social Software systems. However, it is difficult to manage to find relevant content and hard to exchange it between different systems. The Semantically Enabled Social Software perspective tries to overcome these issues. It proposes to augment the existing informal or semiformal structures like hyperlinks with machine-readable formal descriptions ("metadata"). Thus, the meaning behind a connection become explicit. Such metadata allows for more sophisticated services, like improved search and navigation, personalized presentation of content and improved interoperability between systems.

Socially Enabled Semantic Web It makes use of Social Software to create semantic structural metadata. One of the most significant barriers to the adoption of the Semantic Web is the hardness of creating formal, machine-readable content. Creating formal metadata currently requires significant expertise in both, the modeled domain and the used formal languages (e.g., RDF and OWL). The Socially Enabled Semantic Web perspective tries to use the Semantic Social Software as a means to simplify the creation of metadata on the Semantic Web. Semantic Social Software enables the creation of metadata upon existing structures, where hyperlinks reflect real-world relationships that are "natural" to the people using the software, supports the collaboration of people with different backgrounds and expertise, allowing augmenting each other; and provides instant gratification: every bit of formal knowledge contributed by a user is immediately usable.

According to Schaffert, although the two research directions have different goals, they are actually only two sides of the same story, namely "Semantic Social Software" [30]. Although the two perspectives have originally developed separately and with different

[3] http://novaspivack.typepad.com/nova_spivacks_weblog/2004/03/from_applicatio.html

application scenarios in mind, the actual software used in both perspectives shares many properties, even to the extent of being actually the same tool used in different settings.

In the following section we introduce Semantic Wikis as a Semantic Social Software, and discuss them under both, the "Semantically Enabled Social Software" and the "Socially Enabled Semantic Web" perspectives.

6.4.1. Semantic Wikis

Semantic wikis combine wiki properties, such as ease of use, collaboration, and linking, with Semantic Web technologies, such as structured content, knowledge models in the form of ontologies, and reasoning. Semantic wikis connect social and artificial intelligence, supporting users in ways that are not available in traditional wikis.

As we have defined above, a wiki is a Web-based system that enables collaborative editing of Web pages whose most important properties are their openness and flexibility. Recently, wikis are increasingly used as tools to support knowledge management. For example, many companies use wikis to maintain and share knowledge about software projects (source code, documentation, project work plans, bug reports, and so on). Knowledge captured in this fashion is easy to create but increasingly difficult to retrieve. Full-text search functionality is the query mechanism the most wikis support. However, full-text search often is insufficient for retrieving knowledge such as structured data or related pages. To overcome these issues, wikis generally have two ways to make such data more accessible: one is by manually update overview pages that sort pages according to certain criteria (involving a lot of maintenance work), and the other is by providing with additional tools such as categories, extensions for certain metadata types (for example, calendars), and template mechanisms that predefine the structure for certain kinds of pages. Although, category systems tend to be inflexible and take users to learn them.

On the other hand, Semantic Wikis offer a simple formalism for semantically annotating links and wiki articles or other kinds of content. They propose to add metadata to the wiki pages themselves and to the inherent hyperlink structure of a Wiki. For instance, a semantic annotations could describe the meaning of a link in a machine readable fashion. These annotations are useful for improving the display contextual information, for improving navigation by facilitating the access to relevant related information, and for perform "semantic" search by querying the context in addition to the content.

The semantic annotations usually correspond to an ontology that defines the properties that can be associated with different object types. Wiki users edit and maintain the ontology within the semantic-wiki system, using knowledge models that are usually represented in RDF Schema and OWL. The internal representation of annotations with RDF/OWL simplifies the data exchange with other applications.

In addition, semantic wikis also offer a semantic search for querying by semantic relations between objects and possibly an additional automatic or semiautomatic extraction of metadata from wiki articles to simplify the annotation process.

In concordance to the Socially Enabled Semantic Web perspective, Semantic Wikis are also excellent tools for collaborative creation of knowledge models. Based on (existing or emerging) natural language descriptions of concepts and individuals, formal knowledge models can be created. An example of this perspective could be to support a common ontology engineering process by a Semantic Wiki. It could be to start with the writing of a collection of normal (informal) Wiki pages that make up the domain to be modeled, and then to augment the existing hyperlink structure between Wiki pages with machine readable annotations. The first task could be easily achieved by a domain expert, the second task by a knowledge engineer. Both could contribute their expertise and collaborate on the creation of the knowledge model.

Different semantic wiki systems follow different manners to add semantic annotations. In semantic wikis, the ontology is created and maintained by associating each ontology instance and concept with a wiki page. Wiki links and annotations relate the concepts and pages with each other. However, there many different semantic annotation mechanisms: while most provide an extended markup language within the textual editor, some provide the annotation mechanism in a separate annotations editor. The first group follows more the "Semantically Enabled Social Software" approach, where the focus is on the page content and annotations are embedded in the wiki page text (e.g., Semantic MediaWiki[4]). They aim to simplify navigation and collaboration by using semantic annotations. The second group wants to establish wikis as a means to collaboratively create Semantic Web ontologies; being the semantic annotations in the foreground and sometimes even more important than the actual content; they follow the "Socially Enabled Semantic Web" approach (e.g., SweetWiki[5], IkeWiki [29][6], OntoWiki [3][7]).

In the following sections, we discuss both annotation approaches. We will use SemanticMediaWiki and IkeWiki [29] as examples.

6.4.1.1. Annotations in Semantic MediaWiki

The Semantic MediaWiki (SMW) system [39] [40] is an extension MediaWiki[8](a widely used wiki-engine that also powers Wikipedia) and falls in the group of wikis systems that focuses on embedding semantic annotations on the wiki page content.

The Semantic Media Wiki (SMW) is a semantically enhanced wiki engine that enables users to annotate the wiki's contents with explicit, machine-readable information. It collects semantic data that were added to the text of wikipages via a special markup. These special markups are manly to categorize an individual and to define properties of the individuals.

Most of the annotations that occur in the SMW correspond to simple ABox statements in OWL DL, i.e. they describe certain individuals by asserting relations between them, annotating them with data values, or classifying them. The schematic information (TBox) representable in the SMW is intentionally shallow. The wiki is not intended as a general purpose ontology editor, since distributed ontology engineering and large-scale reasoning are currently problematic.

[4]http://semantic-mediawiki.org/wiki/Semantic_MediaWiki
[5]http://sweetwiki.inria.fr/wiki/data/Main/WikiObjectModel.jsp
[6]http://ikewiki.salzburgresearch.at/
[7]http://ontowiki.net/Projects/OntoWiki
[8]http://www.mediawiki.org/wiki/MediaWiki

As in most semantic wikis, in the SMW every article corresponds to exactly one ontological element (including classes and properties), and every annotation in an article makes statements about this single element. Furthermore, all annotations refer to the (abstract) concept represented by a page, not to the HTML document.

Let see an example, the Figure 6.13(a) is a wiki page about Berlin city. Although, it is visualized as any other wiki page, its content it is semantically annotated. Figure 6.13(b) is the previous page in edit mode and shows textual annotations in the SMW. However, we can remark there are many new markups in the text. For instance, the page titled Berlin is an individual that was classified by the `City` category; the annotation

```
[[capital of::Germany]]
```

denotes a `capital` relationship between the concepts `Berlin` and `Germany`; and

```
[[area:=891.69 km² |891.69 square kilometers]]
```

denotes the property `area` which take value `891.69 km`^{2}. This formal representation is now available for querying all cities that are capital of some country.

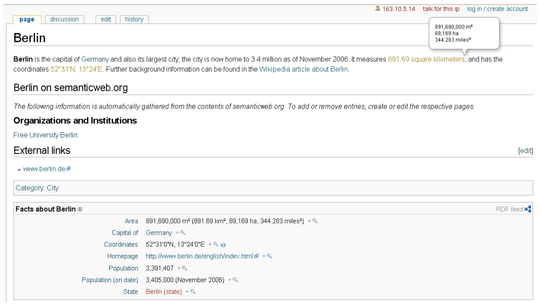
Particularly, properties in the SMW are used to express binary relationships between one semantic entity (as represented by a wiki page) and some other such entity or data value. Each wiki-community is interested in different relationships depending on its topic area, and therefore the SMW allows wiki users control the set of available properties. Thus, the SMW allows links to be characterized by properties, such that the link's target becomes the value of a user-provided property. But unlike RDF-based languages, SMW does not view property statements (subject-predicate-object triples) as primary information units; it rather adopts a page-centric perspective where properties are a means of augmenting a page's contents in a structured way.

Adhering to MediaWiki's basic principles, semantic data in the Semantic Media Wiki (SMW) is also structured by pages, such that all semantic content explicitly belongs to a page. Different namespaces are used to distinguish the semantic roles that wiki pages may play: they can be *individual elements* (the majority of the pages, describing elements of the domain of interest), *categories* (used to classify individual elements, and also to create subcategories), *properties* (relationships between two pages or a page and a data value), and *types* (used to distinguish different kinds of properties).

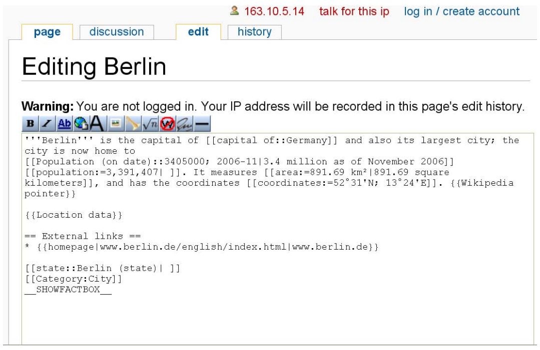
The formal semantics of annotations in the Semantic Media Wiki (SMW) is given via a mapping to the OWL DL ontology language. Most annotations can easily be exported in terms of OWL DL, using the simple mapping from wiki pages to OWL entities: normal pages correspond to abstract individuals, properties correspond to OWL properties, categories correspond to OWL classes, and property values can be abstract individuals or typed literals.

However, it is necessary to achieve to better user interfaces which make easier the semantic annotation editing. For instance, Kaukolu [20] [9] is a wiki system that allows also annotations with extended wiki markup but as well as form-based annotations that are built dynamically from underlying ontologies. Annotations can refer to arbitrary parts

[9] http://kaukoluwiki.opendfki.de



(a) A Semantic Media Wiki (SMW) example. This is the Berlin page.



(b) A Semantic Media Wiki (SMW) example. The edit page of the Berlin page.

of a page rather than just the whole page, and external systems can generate annotations automatically. For example, experiments are currently under way that use eye tracking technology and an eye tracker-based extension to highlight text of existing documents.

6.4.1.2. Annotations in IkeWiki

IkeWiki [29] is another paradigmatic example of the semantic wiki which falls in the group of Socially Enabled Semantic Web. IkeWiki was developed as a tool for knowledge engineers. That is because the purpose of IkeWiki is to be a collaborative and easy to use tool that guides knowledge engineers in the process to make emerging knowledge represented in a formal language or ontology from informal content of the wikipages.

A typical workflow in IkeWiki could be to start with a set of Wiki pages from Wikipedia and modify them according to one's needs and requirements. Then, a knowledge engineer could gradually associate types with the pages and the links between pages, thus increasingly formalizing the knowledge represented in the Wiki. Finally, he could download the Wiki contents as a RDF or an OWL ontology for further use in other applications.

Figure 6.13 shows form-based annotation in the IkeWiki system [29]. The annotation is for the term "Bilberry" in a biology application. The system allows users to create and to annotate not only links and pages, but also ontology classes and properties. It includes several kinds of semantic metadata: (1) Type information is shown below the page title, (2) Links to (semantically) related pages are displayed in a separate "references box" on the right hand side, and finally, (3) shows interactive typing of links using AJAX technology, making it simple for users to add metadata to existing structures.

IkeWiki can be seen as ontology editing tool like a "RDF graph editor". It is fully capable of editing everything that can be represented in RDF, even OWL ontologies. IkeWiki provides basic support for common ontology editing tasks like creating and editing classes and properties, defining range and domain for properties, defining superclasses, etc. More sophisticated ontology features can be loaded into IkeWiki but need to be edited either externally or at the RDF level.

IkeWiki's "first class citizens" are RDF Resources. Each "article" or "wiki page" corresponds `1:1` to a resource in the knowledge base. Meanwhile, an article can be seen as a human readable description of the corresponding resource, the resource's context in the knowledge base can be seen as the machine readable description of that resource.

IkeWiki's most fundamental support for Semantic Web technologies is the annotation of pages and links with concepts / properties from the ontologies that are represented in the knowledge base. Intuitively, page annotation means that we tell IkeWiki what the current resource is about. For example, the description of a person could be annotated with `foaf:Person`. Therefore, page annotation means that a class (OWL or RDFS) is associated with the currently selected RDF resource/article (when adding type T to resource R, IkeWiki adds a triple (`R rdf:type T`) to the RDF knowledge base). By default, every resource is automatically annotated with type `rdfs:Resource`.

As articles correspond to resources, links between articles correspond to relations between resources. The underlying meaning of a link is made explicit by annotating the link with property types (OWL ObjectProperties) from the knowledge base. For example, the link from the article describing the country `Germany` to the city `Berlin` can be annotated with the property type `hasCapital`. Notice that, relations between resources are independent from the actual navigational links existing in the content of the wikipage.

For each article, IkeWiki already sets a number of default relations to other resources, e.g. a `ikewiki:hasAuthor` relation to the author of the wikipage.

Page and link annotation is very easy in IkeWiki. IkeWiki use the "+" to open a "contextualized" dialogue to select a type from the list of OWL and RDFS classes or a property types to be added that are currently stored in the knowledge base.

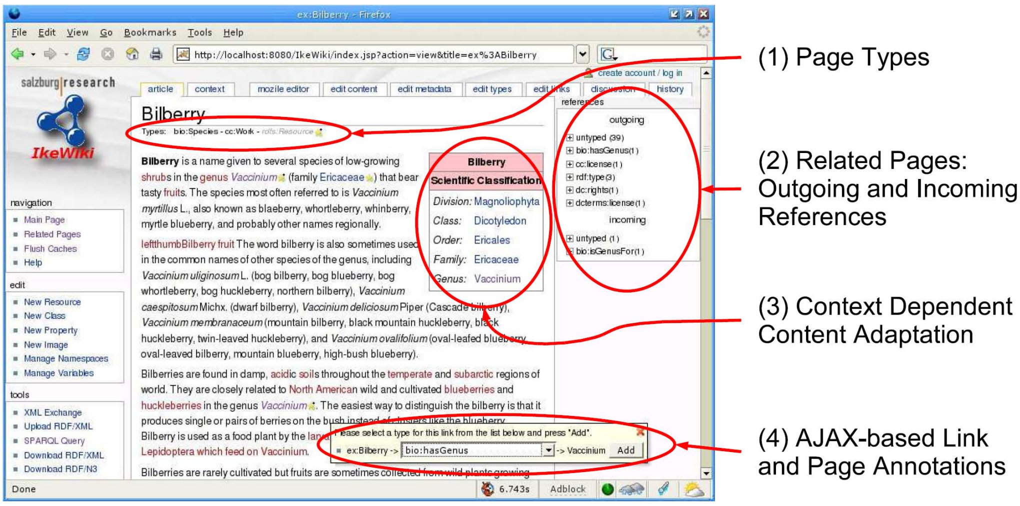


Figure 6.13. An IkeWiki example

6.4.1.3. Annotations in others Semantic Wiki Systems: SweetWiki and OntoWiki

SweetWiki [9] is a research prototype from Inria Sophia-Antipolis[10] which combines social tagging with formal ontologies. Users can easily annotate pages with arbitrary tags, which they can in turn associate with concepts from the underlying ontologies. Users can tag pages, pictures, attached files, etc.

SweetWilki also proposes a mixed approach in order to "organize the tags". It link the tags together within a folksonomy described using the semantic web languages, where tags are organized in a hierarchy and related to one another using relationships like subClassOf, seeAlso, etc.

On the other hand, OntoWiki [3] system differs from the others in that classical textual content is no longer in the foreground. Instead, OntoWiki offers an easy-to-use interface for collaboratively creating and maintaining ontologies. Based on an imported ontology (RDF or OWL), users can populate the ontology guided by the ontology. It also supports semantic search and navigation as well as the possibility of versioning metadata.

6.4.2. Exploiting Semantics

In this section we discuss the usefulness of semantic markup.

6.4.2.1. Browsing

In Figure 6.13(a), the page includes a so called Factbox which is placed at the bottom of the page. In the Semantic Media Wiki, the Factbox summarizes the given semantic annotations, provides feedback on possible errors, and offers links to related functions. These links can be used to browse the wiki based on its semantic content. The page title in the Factbox heading leads to a semantic browsing interface (Figure 6.14) that shows not only the annotations within the given page, but also all annotations where the given page is used as a value. The magnifier icon behind each value leads to an inverse search for all pages with similar annotations. In addition, the Factbox shows links to property pages, which in turn list all annotations for a given property. All those browsing features are interconnected by appropriate links, so that users can easily navigate within the semantic knowledge base.

6.4.2.2. Querying

The Semantic Media Wiki (SMW) includes a query language that allows access to the wiki's knowledge. The query language can be used in two ways: either to directly query the wiki, or to add the answer to a page by creating an inline query. The latter enables editors to add dynamically created lists or tables to a wikipage, thus keeping up-to-date the wikipage.

Figure 6.15 shows an example query for all cities that are located in an EU-country or that have more than 500,000 inhabitants:

[10]http://www-sop.inria.fr/teams/edelweiss/wiki/wakka.php?wiki=SweetWiki

Figure 6.14. Semantic Browsing Interface in the Semantic Media Wiki (SMW)

.

```
[[Category:City]]
  <q>[[located in::
     <q>[[Category:Country]] [[member of::EU]]
     </q>]]
     ||
     [[population:: >500,000]]
  </q>
```

Figure 6.15. A Semantic Media Wiki (SMW) example query

In the Semantic Media Wiki, the result of SMW-queries is a set of pages. In order to retrieve more information about those results, SMW allows so-called print requests as parts of queries. For instance, writing `[[hasCapital::C]]` within a query will cause all values of the property `has capital` to be displayed for each result.

On the other hand, IkeWiki allows to query the RDF knowledge base using the SPARQL[11] query language from inside wiki pages. Such wiki pages are then dynamically rendered, filling in the result values of the SPARQL query. IkeWiki supports the full SPARQL SELECT syntax, including filtering, regular expressions, etc. Note however, that using SPARQL requires extensive knowledge of RDF and the data model stored by IkeWiki.

6.4.3. Salient Aspects of Semantic Social Software

Semantic Social Software has a number of properties that make it interesting as a research topic besides the two major "sides" described above. Shaffert [30] presents an overview over what he considers salient aspects of Semantic Social Software, which then we have adapted to semantic wikis:

[11] http://www.w3.org/TR/rdf-sparql-query/

Testbed for Semantic Web Technology Semantic Wikis as any Semantic Social Software is software that can be developed quickly and easily, and can build upon existing applications and principles. At the same time, semantic wikis shows many of the promises and also of the problems of the big "Semantic Web Vision." Examples are the improved searching and navigation, personalization and content adaptation, interoperability, open world assumption, coupling of data and metadata, evolving knowledge models, inconsistencies in real-world data, ontology alignment, and so forth. If technology works properly in Semantic Wikis, there is also a high chance that it works on the large Semantic Web.

Coupling of Data and Metadata In Semantic wikis, the combination of data and metadata is quite natural, as existing Social Software with existing content is merely augmented (and not replaced) by metadata. A combination of data and metadata requires consideration of query and reasoning languages that are capable of processing both, data and metadata. A further issue that often becomes apparent in Semantic Wiki is the so-called URI crisis, that is, the question whether a URI used in the metadata refers to a Wiki page or to the content described in that page. Another interesting aspect of coupling data and metadata is the semi-automatic extraction of metadata from the actual data using, for example, natural language processing techniques. As automatic metadata extraction is currently not perfectly reliable, it would make sense to mark such annotations as "automatic" and give users the opportunity to revise them if needed.

emerging and evolving knowledge models A knowledge model in a Semantic Wiki may begin with a small set of annotations for a single page to a full-fledged ontology. Such evolving knowledge models raise a number of interesting questions that are also relevant for the Semantic Web: primarily, evolving systems will be full of inaccuracies and even inconsistencies, demanding for more tolerant formal languages than those that are proposed today; also, trust, versioning, and merging/aligning are issues that will need to be addressed appropriately.

Self-Organizing Communities around Emerging Topics In Semantic Wikis, emerging topics could be identified automatically by applying appropriate reasoning. Such information could be used to provide readers with information about "what is relevant".

6.5. Anatomy of a Semantic Wiki Engine

A semantic wiki engine is basically a traditional wiki engine extended with capabilities to handle semantic features: semantic annotation creation, storage and management, semantic querying and reasoning. It is worth noting that Semantic Media Wiki (SMW), one of the main semantic wiki engine and probably the most used actually is built and distributed as an extension to MediaWiki, a traditional wiki engine. In this section, we will first describe the architecture of a traditional wiki engine. Then we will present the different components of a semantic extension for a wiki engine, and discuss the ways to implement them, based on the examples of two semantic wiki engines, Semantic Media Wiki and IkeWiki.

6.5.1. Architecture of a traditional Wiki Engine

A wiki engine is a web application that offers services to create, edit and navigate wiki pages through a standard web browser. These pages are described using a particular syntax, called the "Wiki Syntax". This syntax allows to format the text and to insert links between pages in a simplified manner compared to HTML. Pages are stored on the server side and are navigated like any HTML page on a website.

A simplified architecture of a traditional wiki engine is presented in Figure 6.16. Three main components are part of a wiki engine:

- the web application (WebApp) itself, in charge of handling user requests,
- the storage component, in charge of the persistence of the wiki pages,
- the wiki renderer, in charge of translating the wiki syntax to standard HTML to produce an output readable by a web browser.

6.5.1.1. The wiki renderer

The renderer is the component in charge of producing plain HTML from text formatted using the wiki syntax. It takes as input the textual content of a page and outputs the same content formatted using HTML.

As an example, using the MediaWiki syntax, the following page fragment:

```
== Anatomy of a Semantic Wiki ==
```

is translated by the MediaWiki renderer to:

```
<h2> <span class="mw-headline">
Anatomy of a Semantic Wiki </span></h2>
```

that produces the following output in the client browser :

```
Anatomy of a Semantic Wiki
```

A wiki renderer is a quite simple procedure based on `regexp` algorithms that looks for predefined patterns in the input text (e.g. `== (.*) ==`) and replace them by the corresponding HTML tags (e.g. `<h2><span ...> (.*)</span></h2>`) in the output text.

It is worth noting that there is actually no agreement on a standard wiki notation: there are multiple flavors of the wiki syntax. Many wiki renderers are thus dedicated to a particular flavor. This is for example the case of the MediaWiki renderer that handles only the MediaWiki syntax. However, some wiki engines use a more advanced renderer capable of handling multiple syntax flavors. This is for example the case of Xwiki[12], an open-source wiki engine that uses the WikiCreole syntax [13], the MediaWiki syntax and many others. Finally, it is interesting to quote some reusable rendering tools like Radeox [19], that allows to integrate basic wiki features in any system.

[12] http://www.xwiki.com/

[13] http://www.wikicreole.org/

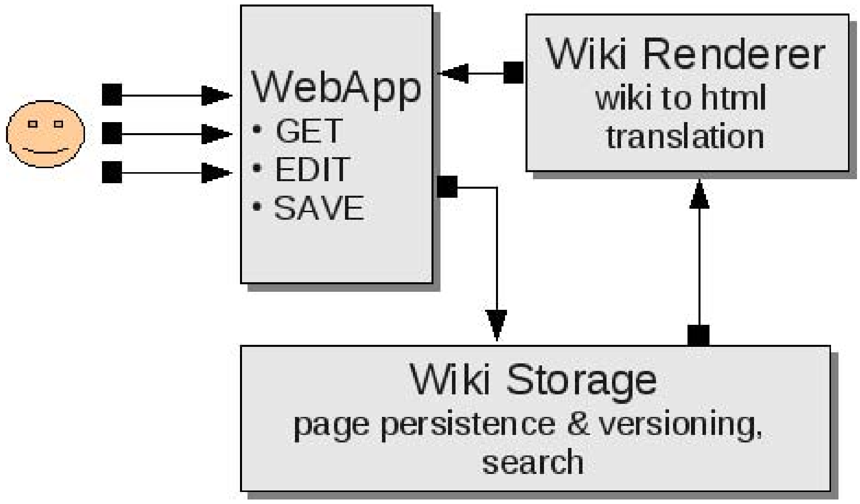


Figure 6.16. Wiki Architecture

6.5.1.2. The storage component

The storage component provides a persistence service for wiki pages. In addition, since one of the main features of a wiki is to keep track of any change done to any page, the storage service must store all the successive versions of all the pages. Thus, all page revisions are explicitly stored as a separate entry in the storage system and can be retrieved.

There is basically two approaches to implement the storage component: files and database.

File based storage consists in simply storing page revisions in files. File names are obviously chosen to be easily computed from the page id. Thus, when a particular page is requested by a user, the corresponding file is accessed using the primitives of the underlying file system. This approach is interesting for simplicity reasons since it does not require the installation of a database engine. On the contrary, it suffers of some limitations: searching the wiki content can be difficult, particularly if the query involves several pages, additional metadata (e.g. author, creation date, comments, attachments ...) are difficult to store and search. The use of a revision control system (e.g. RCS) is an interesting solution and offers a efficient file storage, but imposes severe difficulties for full text search. Some wikis use a file storage, the main one being pmWiki[14] .

Database storage is the most commonly used approach. It consists in using a database engine to stoge pages, page revisions and page contents. To represent the revision history, pages, revision and text content are stored in different tables. A simplified representation of the storage model used by MediaWiki is given in Figure 6.17.

The main interests of a database approach for the wiki storage component rely on the features of the database engine: powerful search thanks to SQL, extensibility and natural support to page metadata by simply adding columns to tables, reliability and security

[14]http://www.pmwiki.org/

through the transaction system. The drawback is that installing such a wiki requires the installation, and administration of a database engine.

A large majority of wiki engines are based on the use of a database to store their content. This is particularly the case for MediaWiki, for many engines dedicated to corporate solutions (Xwiki, Confluence[15], etc.), and for most of the semantic wikis (SMW, IkeWiki). Most of them uses a open-source database (MySql, Postgres).

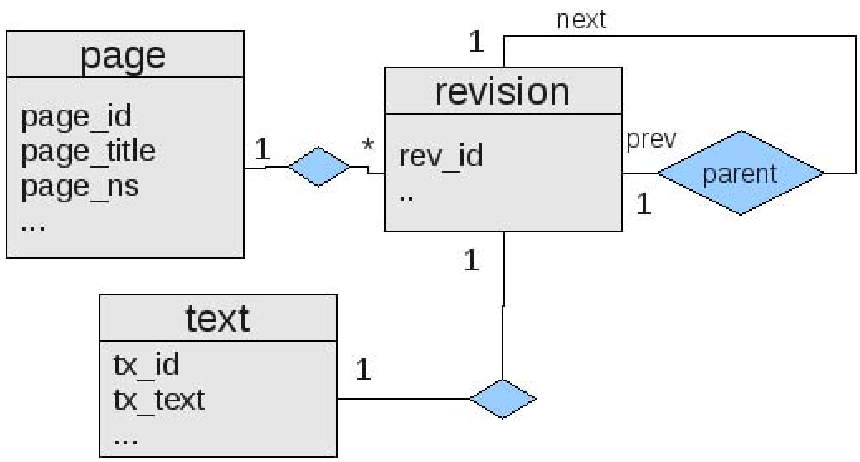


Figure 6.17. A simplified representation of the storage model used by MediaWiki

6.5.1.3. The WebApp

The wiki web application (WebApp) is the component in charge of interacting with the client and handling user requests. There is three requests:

GET (pageId) correspond to a navigation request, where the user asks for a particular page to be displayed. The handling of this request consists in

- retrieve the latest revision of the required page from the storage component,
- pass it to the renderer to get the HTML version of the content,
- return the HTML text back to the client.

EDIT (pageId) correspond to the action of a client asking for entering the editing mode for a given page. The handling of this requests simply consists in retrieving the content of the latest revision of the required page from the storage component, and returning it back to the client without converting it to HTML. This content is inserted in an HTML TEXTAREA element to be editable through the client browser.

SAVE (pageId, newContent) correspond to the action of a user requesting for finishing a edit session by saving the changes done to a page content. The handling of this request consists in creating a new revision for the page being saved, with the content

[15] http://www.atlassian.com/software/confluence/

provided the client as the text content associated to the new revision. The parent revision of the new one is set to the current revision at the time of the SAVE.

The wiki WebApp component is built using standard web development technologies: Java servlets or PHP scripts in the most common cases.

6.5.2. Architecture of a Semantic Wiki Engine

As introduced in the previous sections, a semantic wiki adds semantic annotations to the textual content of the wiki page. These annotations map the wiki content to a formal structure and offer for semantic search and reasoning facilities over the wiki content.

In a large majority of actual semantic wiki engines, the annotation subject is always a wiki page. The annotation object can be taken from a preloaded ontology, from a simple vocabulary, or can be a wiki page. In any cases, annotations are triples (subject, relation, object) and express facts about the wiki content. An example of an annotated wiki content is given in Figure 6.18.

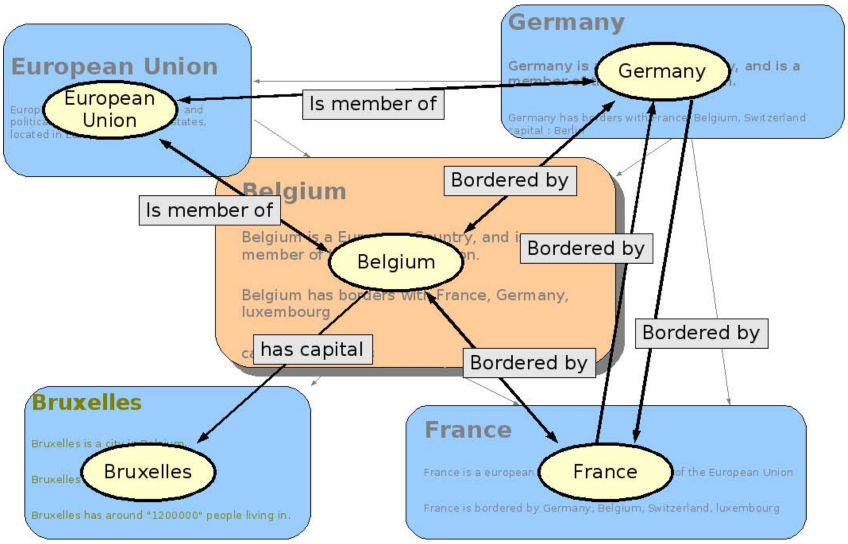


Figure 6.18. A Semantic wiki: pages + formal annotations

In this example, the `Belgium` page is the subject of several annotations, each of the annotations represented by a triple:

(`Belgium`, `Bordered_by`, `France`),

(`Belgium`, `Bordered_by`, `Germany`),

(`Belgium`, `has_capital`, `Bruxelles`) and

(`Belgium`, `member_of`, `European Union`).

From this set of facts, it is easy to express a query that retrieve all countries having a common border with Belgium.

From an architectural point of view, a semantic wiki engine appears to be simply a classical wiki engine extended with triple management facilities for their creation, storage, querying and in some cases reasoning.

6.5.2.1. Storage component for a semantic wiki

The storage component of a semantic wiki provides a persistence service for wiki pages and triples representing annotations. In some wikis, textual content of pages and annotations are created separately, while in some others, annotations are embedded inside the textual content of pages. However, in any case, text and triples are stored separately, like in the example illustrated in Figure 6.19. In other words, the storage component of a semantic wiki is built by extending a traditional wiki storage component with a triple store.

Note that, in wikis in which annotations are embedded inside the textual content of the pages (e.g. Semantic MediaWiki), these annotations need to be extracted from the textual content each time a page is saved to be stored in the triple store, while in wikis in which annotations are created separately, the annotations can be directly stored in the triple store.

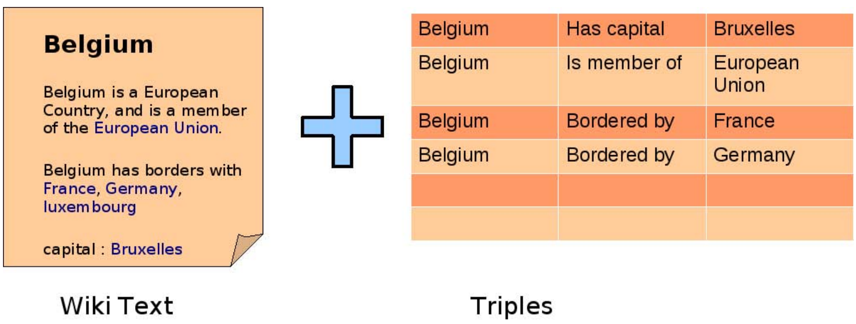


Figure 6.19. Storage of the "Belgium" page of the example

There are basically two solutions to implement the triple store. The first one is to use an existing RDF store engine, like JENA[16] or SESAME[17]. This approach is interesting because these engines offer numerous and powerful facilities to handle RDF triples: persistence, but also SPARQL queries, inference reasoning and various API for RDF, RDFS and OWL management. The main drawback of this kind of solution relies to performance issues. The problem is that while pages and triples are stored in two different engines, evaluating a semantic query always requires the evaluation of at least two queries: one against the triple store, and the other against the wiki database. This solution is the one adopted by IkeWiki.

The second solution is to implement the triple storage by some dedicated tables in the wiki database. This solution is better for performance, but requires to implement specific procedures for querying the triples. This solution is the one adopted by Semantic MediaWiki.

[16] http://jena.sourceforge.net/
[17] http://www.openrdf.org/

6.5.2.2. Annotation Management

From the user point of view, there is two ways of annotating wiki pages. In wikis like IkeWiki, annotation is a separate activity from page content creation. On the other hand, in wikis like SMW, annotations are embedded in the text content and are thus created as part of the page editing activity.

In the first case, the wiki interface provides explicit means to create, update or delete annotations. The user actions related to annotation management are expressed by specific requests (i.e. CREATE_ANNOTATION, DEL_ANNOTATION, UPDATE_ANNOTATION) that must be handled by the wiki WebApp. The WebApp handles them by directly interacting with the triple store component.

In the second case, users create annotations by simply inserting them within the text content of the pages, using a some dedicated extensions to the wiki syntax. This approach, compared to the previous one, does not requires to extend the wiki WebApp. However, annotations need to be extracted from the text content at the save time.

6.5.2.3. Handling Semantic Queries

Semantic queries are queries in which at least a part need to be evaluated against the triple store. Semantic queries are handled by a specific engine capable of querying the triple store and the page storage before and combining them to form the result. The result of a semantic query can be an existing page extracted from the page storage, a list of existing pages, or a newly created page (i.e. a page whose content has been computed by a querying/inferencing algorithm and not edited by a user). Depending on the query engine, some reasoning can be done while querying (e.g. inferencing, subsumption). In particular, some query engine are capable of taking into account RDFS or OWL descriptions.

The same dichotomy that exists for annotations creation also exists for queries. Wikis in which annotations are separated from pages adopt the same approach for queries by offering specific means for expressing semantic queries in their interface. Wikis that use embedded annotations also use embedded queries inside page contents.

The first case is implemented in the same way than annotation management: the WebApp is extended to handle requests corresponding to semantic queries. These queries can be directly passed to the semantic query engine.

In the second case, the WebApp do not need to be extended. Queries are executed when a user requests the access (i.e. GET request) to a page whose content contains a semantic query. When the page content is retrieved from the page storage, the semantic queries are extracted from that text, passed to the query engine to be evaluated. These queries are then replaced by their result inside the text. Finally, the new page content, that embed the results of the semantic queries, is sent to the renderer.

6.5.3. The Big Picture

To summarize the previous section, we return to the simplified architecture of a wiki to extend it with the components of a semantic extension. There is two different cases: one that correspond to wikis in which annotations and queries are separated from the pages, and the other that correspond to wikis in which annotations and queries are embedded inside page contents. Both approaches require to introduce a triple store component, that can be implemented as an extension of the wiki database or by using a specialized engine, and a semantic query engine that evaluates queries against the triple store and against the page store. In the first class of wikis referred as "IkeWiki-like wikis", whose architecture is illustrated in Figure 6.20, the main extension resides in new requests handled by the WebApp.

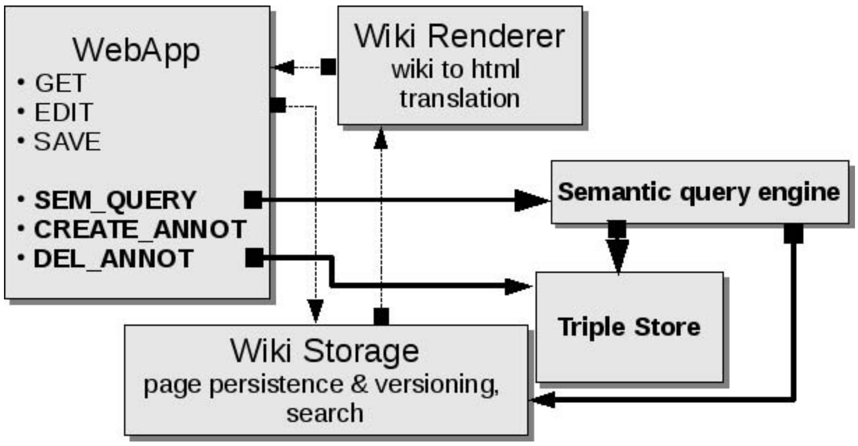


Figure 6.20. Architecture of an "IkeWiki-like" semantic wiki

On the other hand, in the second class of wikis, referred as "SMW-like wikis", the main extension resides in the insertion of annotations and query extractor from the page contents. This class of wikis is illustrated in Figure 6.21.

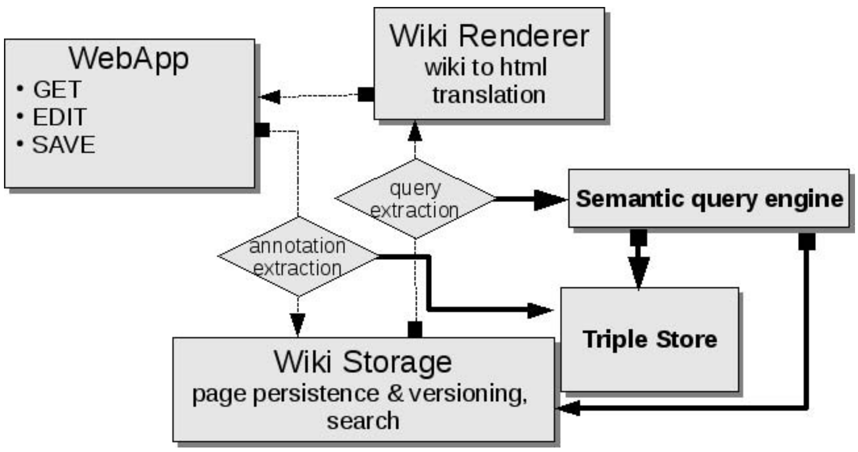


Figure 6.21. Architecture of a "SMW-like" semantic wiki

6.6. Research directions

In this section we point to research reports which are illustrative of recent efforts carried out in the themes of knowledge discovery, ontologies and semantic wikis.

Knowledge representation and discovery has been the focus of several research efforts. Representative results report research relative to design new semantic wikis [4], the identification of clusters based on semantic wikis [32], ontology extraction from knowledge bases[11], and metric-based ontology learning [43].

Research involving the construction of ontologies includes support to the semi-automatic construction of ontologies using ontology patterns [7], to the alignment of ontologies using design patterns [31], to the mapping of ontologies using algorithms that acquire relationships between ontological components [36], and to the identification of concepts in large text (email) collections using text mining, information retrieval, natural language processing and machine learning techniques [44]. The cost of reusing ontologies has been tackled, for instance, by the proposal of a framework for selecting the most appropriate ontology for mining biomedical text [38].

Recent research involving semantic wikis includes investigations relative to support to searching [14] in general and with focus in facilitating searches by end-users in particular [2]. Several efforts have also been targeted at applying the concepts to business process [17] and workflows [16].

Acknowledgments

We thank the supporters of the STIC-Amsud Program, which include CNRS and INRIA in France, FAPESP and CAPES in Brazil, SECYT in Argentina, and CONICYT in Chile.

References

[1] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer (Cooperative Information Systems)*. The MIT Press, April 2004.

[2] Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Paslaru Bontas Simperl, editors. *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings*, volume 5554 of *Lecture Notes in Computer Science*. Springer, 2009.

[3] Sören Auer, Sebastian Dietzold, and Thomas Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In Isabel F. Cruz, Stefan Decker, Dean Allemang, Chris Preist, Daniel Schwabe, Peter Mika, Michael Uschold, and Lora Aroyo, editors, *The Semantic Web - ISWC 2006, 5th International Semantic*, volume 4273 of *Lecture Notes in Computer Science*, pages 736–749. Springer, 2006.

[4] Joshua Bacher, Robert Hoehndorf, and Janet Kelso. Bowiki: Ontology-based semantic wiki with abox reasoning. In *Proceedings of the 3rd Semantic Wiki Workshop (SemWiki 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*, 2008.

[5] Rokia Bendaoud, Amedeo Napoli, and Yannick Toussaint. Formal concept analysis: A unified framework for building and refining ontologies. In *Knowledge Engineer-*

ing: Practice and Patterns, 16th International Conference, EKAW 2008, Acitrezza, Italy, September 29 - October 2, 2008. Proceedings, pages 156–171, 2008.

[6] Rokia Bendaoud, Yannick Toussaint, and Amedeo Napoli. Pactole: A methodology and a system for semi-automatically enriching an ontology from a collection of texts. In *Conceptual Structures: Knowledge Visualization and Reasoning, 16th International Conference on Conceptual Structures, ICCS 2008, Toulouse, France, July 7-11, 2008, Proceedings*, pages 203–216, 2008.

[7] Eva Blomqvist. Pattern ranking for semi-automatic ontology construction. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 2248–2255, New York, NY, USA, 2008. ACM.

[8] Uwe M. Borghoff and Johann H. Schlichter. *Computer-Supported Cooperative Work: Introduction to Distributed Applications*. Springer, 2000.

[9] Michel Buffa, Guillaume Erétéo, Catherine Faron-Zucker, Fabien Gandon, and Peter Sander. SweetWiki: A Semantic Wiki. *Journal of Web Semantics, special issue on Web 2.0 and the Semantic Web*, 6(1), february 2008.

[10] Claudio Carpineto, Stanislaw Osiński, Giovanni Romano, and Dawid Weiss. A survey of web clustering engines. *ACM Comput. Surv.*, 41(3):1–38, 2009.

[11] Adrien Coulet, Malika Smaïl-Tabbone, Amedeo Napoli, and Marie-Dominique Devignes. Ontology refinement through role assertion analysis: Example in pharmacogenomics. In Franz Baader, Carsten Lutz, and Boris Motik, editors, *Description Logics*, volume 353 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[12] Piero Fraternali. Tools and approaches for developing data-intensive web applications: a survey. *ACM Comput. Surv.*, 31(3):227–263, 1999.

[13] Irene Greif and Sunil K. Sarin. Data sharing in group work. *ACM Trans. Inf. Syst.*, 5(2):187–211, 1987.

[14] Peter Haase, Daniel Herzig, Mark A. Musen, and Thanh Tran. Semantic wiki search. In *The Semantic Web: Research and Applications, 6th European Semantic Web Conference, ESWC 2009, Heraklion, Crete, Greece, May 31-June 4, 2009, Proceedings (LNCS 5554)*, pages 445–460, 2009.

[15] Siegfried Handschuh and Steffen Staab. Authoring and annotation of web pages in cream. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 462–473, New York, NY, USA, 2002. ACM.

[16] Gobe Hobona, David Fairbairn, and Philip James. Semantically-assisted geospatial workflow design. In *GIS '07: Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, pages 1–8, New York, NY, USA, 2007. ACM.

[17] Toufeeq Hussain, Rajesh Balakrishnan, and Amar Viswanathan. Semantic wiki aided business process specification. In *Proceedings of the 18th International Conference on World Wide Web, WWW 2009, Madrid, Spain, April 20-24, 2009*, pages 1135–1136, 2009.

[18] Melody Y. Ivory and Marti A Hearst. The state of the art in automating usability evaluation of user interfaces. *ACM Comput. Surv.*, 33(4):470–516, 2001.

[19] Matthias L. Jugel and Stephan J. Schmidt. The radeox wiki render engine. In Dirk Riehle and James Noble, editors, *Proceedings of the 2006 International Symposium on Wikis, 2006, Odense, Denmark, August 21-23, 2006*, pages 33–36. ACM, 2006.

[20] Malte Kiesel. Kaukolu: Hub of the semantic corporate intranet. In Max Völkel and Sebastian Schaffert, editors, *SemWiki2006, First Workshop on Semantic Wikis - From Wiki to Semantics, Proceedings, co-located with the ESWC2006, Budva, Montenegro, June 12, 2006*, volume 206 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.

[21] Mei Kobayashi and Koichi Takeda. Information retrieval on the web. *ACM Comput. Surv.*, 32(2):144–173, 2000.

[22] Markus Krötzsch, Denny Vrandecic, and Max Völkel. Semantic mediawiki. In *The Semantic Web - ISWC 2006, 5th International Semantic Web Conference, ISWC 2006, Athens, GA, USA, November 5-9, 2006, Proceedings (LNCS 4273)*, pages 935–942, 2006.

[23] Bo Leuf and Ward Cunningham. *The Wiki Way: Quick Collaboration on the Web*. Addison-Wesley Longman, Amsterdam, pap/cdr edition, 2001.

[24] Louvre Museum, Paris. http://www.louvre.fr/.

[25] Art Museum of Sao Paulo. http://masp.art.br/.

[26] Deborah L. Mcguinness and Frank van Harmelen. OWL web ontology language overview. W3C recommendation, W3C, February 2004.

[27] Weiyi Meng, Clement Yu, and King-Lup Liu. Building efficient and effective metasearch engines. *ACM Comput. Surv.*, 34(1):48–89, 2002.

[28] Metropolitan Museum of Art, New York. http://www.metmuseum.org/.

[29] S. Schaffert. IkeWiki: A Semantic Wiki for Collaborative Knowledge Management. *1st International Workshop on Semantic Technologies in Collaborative Applications (STICAÇ06), Manchester, UK*, 2006.

[30] Sebastian Schaffert. *Emerging Technologies for Semantic Work Environments: Techniques, Methods, and Applications*, chapter Semantic Social Software: Semantically Enabled Social Software or Socially Enabled Semantic Web?, pages 33–45. J. Rech, B. Decker and E. Ras Eds. Information Science Reference (IGI Global, April 2008.

[31] François Scharffe, Jérôme Euzenat, and Dieter Fensel. Towards design patterns for ontology alignment. In *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, pages 2321–2325, New York, NY, USA, 2008. ACM.

[32] Gero Scholz. Semantic mediawiki with property clusters. In *Proceedings of the 3rd Semantic Wiki Workshop (SemWiki 2008) at the 5th European Semantic Web Conference (ESWC 2008), Tenerife, Spain, June 2nd, 2008*, 2008.

[33] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, 2002.

[34] Semantic Media Wiki. http://semantic-mediawiki.org/wiki/Semantic_MediaWiki.

[35] S. Staab and R. Studer. *Handbook on Ontologies*. Springer, 2004.

[36] Suzette Kruger Stoutenburg. Acquiring advanced properties in ontology mapping. In *PIKM '08: Proceeding of the 2nd PhD workshop on Information and knowledge management*, pages 9–16, New York, NY, USA, 2008. ACM.

[37] Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 697–706, New York, NY, USA, 2007. ACM.

[38] He Tan and Patrick Lambrix. Selecting an ontology for biomedical text mining. In *BioNLP '09: Proceedings of the Workshop on BioNLP*, pages 55–62, Morristown, NJ, USA, 2009. Association for Computational Linguistics.

[39] Max Völkel, Markus Krtözsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. *Journal of Web Semantics*, 5(4), 2007.

[40] Max Völkel, Markus Krötzsch, Denny Vrandecic, Heiko Haller, and Rudi Studer. Semantic wikipedia. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 585–594, New York, NY, USA, 2006. ACM.

[41] James Whitescarver, Prithviraj Mukherji, Murray Turoff, Ronald J. DeBlock Jr., Robert M. Czech, and Bijoy K. Paul. A network environment for computer-supported cooperative work. *Computer Communication Review*, 17(5):260–272, 1987.

[42] Wikipedia. http://en.wikipedia.com/.

[43] Hui Yang and Jamie Callan. Metric-based ontology learning. In *ONISW '08: Proceeding of the 2nd international workshop on Ontologies and nformation systems for the semantic web*, pages 1–8, New York, NY, USA, 2008. ACM.

[44] Hui Yang and Jamie Callan. Ontology generation for large email collections. In *dg.o '08: Proceedings of the 2008 international conference on Digital government research*, pages 254–261. Digital Government Society of North America, 2008.